OS-9 USERS GROUP
"DEDICATED TO EXCELLENCE IN OS-9 COMPUTING"

M.O.T.D.

The OS-9 Users Group Newsletter

Issue #4, 1995

About the MOTD

MOTD Information

The OS-9 Users Group, Inc. is a not-for-profit organization, registered and incorporated in the state of Iowa, whose members share an interest in the OS-9 operating system in all of its various forms.

MOTD is the official newsletter of the *OS-9 Users Group, Inc.*

The OS-9 Users Group, Inc. has no affiliation with ***Microware*** or any other organization.

The opinions expressed by the authors of any articles or columns are not necessarily the opinions of the Editor, Publisher, columnist, nor do they reflect the policies of *The OS-9 Users Group, Inc.*

The **MOTD** will be printed a minimum of four times per year.

To receive **MOTD** you must be a member in good standing of *The OS-9 Users Group, Inc.* in accordance with the bylaws.

To become a member in good standing you must pay the yearly dues of $25.00 US funds ($30.00 US funds if you live outside the US and Canada) and obey all of the membership rules set forth in the Constitution of The Users Group and the bylaws.

Dues should be made payable to:

The OS-9 Users Group, Inc.
6158 West 63rd Street
Suite 109
Chicago, Il. 60638
U.S.A.

Memberships run for one year from the date received by the Users Group. Send a SASE for a membership kit which will include a membership form and information about the Users Group.

Please remember that this is a self help organization, and also a non-profit organization incorporated in the state of Iowa and as such it is directly governed by the laws of that state as well as all federal laws. Accordingly, dues are non-refundable in accordance with both state and federal law.

Reprints or back issues of the **MOTD** are available to members in good standing at the cost of $1.50 each plus $0.50 shipping (US funds). Please send a SASE and a list of the issues you wish sent to you.

Here is a list of the current *OS-9 Users Group* officers:

Position	Name
President	Colin McKay
Executive Vice Pres.	David Graham
Director	Ed Gresick
Director	Eddie Kuns
Director	Ken Scales
Vice Pres. (Comm.)	Paul Jerkatis
Vice Pres. (OS-9)	Brian Goers
Vice Pres. (OS-9/6809)	Mike Rowen
Secretary	Howard Luckey
Treasurer	Br. Jeremy
Librarian	Dave Kelly
MOTD Editor	Joel Mathew Hegberg

MOTD Contributions

Articles, editorials, "letters to the Editor/Board of Directors", personal ads, graphics, or columns may be submitted by using the following means:

E-mail to JoelHegberg@delphi.com, E-mail to 'Sysop' of SandV BBS [(708) 352-0948], E-mail to the *OS-9 Users Group* at os9ug@sandv.chi.il.us or by mail to *The OS-9 Users Group* address in Chicago. All submissions should be in pure ASCII format.

The submission of material does not guarantee publication. All publication of material is subject to review by the Board of Directors and the **MOTD** Editor, and must not be in conflict with the stated purposes of the Users Group as defined by the constitution and bylaws of the Users Group. The Board of Directors may also establish additional guidelines for acceptance to publication. Submission deadlines are four weeks before actual printing of the **MOTD** issue. *All printed material is subject to nominal editing by the **MOTD** Editor for clarity.*

Suggestions for making **MOTD** a better publication are welcome. Letters may be addressed to the **MOTD** Editor and mailed to the Users Group address or sent directly to JoelHegberg@delphi.com. E-Mail can also be sent to the following Delphi members: NIMITZ, MITHELEN, LUCKYONE, BRIANGOERS, EDDIEKUNS, EDELMAR, MROWEN, REVWCP, and KSCALES. You may also sent E-mail to os9ug@sandv.chi.il.us.

MOTD Advertising

Commercial advertising is available in **MOTD**. Please send a SASE for current rates. All ads should be submitted as a camera-ready copy. We reserve the right to limit the size and quantity of ads.

SandV BBS (708) 352-0948

The SandV BBS is a BBS being maintained by Paul Jerkatis to provide Internet access.

From the President

First, thanks to all of you for the feedback about the 9503 issue. A few delays in getting it out, but I think that the end results were worthwhile. This issue gets us almost back on schedule for the year, and the next two issues will follow in October (with an Atlanta Fest report), and in early December. Everybody liked the new format, though there were a few problems with photocopying dark areas which we hope to improve upon. That issue was mailed from Canada, but now that Joel has completed his move mailing from the US (cheaper postage) has resumed.

Atlanta Fest

As mentioned in the last issue, the Atlanta Fest is approaching quickly, 30 September to 1 October! See the ad below for more details. Should be lots of interesting stuff to see there, along with the laser light show at Stone Mountain, and a chance to meet fellow OS-9 enthusiasts.

UG Library

The OS-9 Community Network should have the complete UG library by the time you read this. The files will be reorganized in accordance with OCN's filing system. See the OCN article in the 9503 issue for details. Once things get organized, we will begin publishing the complete OCN library listing in the MOTD over a period of several months.

MIT Sea Grant College Program

In the September 1995 issue of Popular Science, on page 26, is an article about a Robot Lobster which is used to follow chemical plumes. Those of you who follow the SFTHM column may recall that the lobster was mentioned in the May/June 1994 issue of the MOTD, in the SFTHM column about the Sea Grant Program's Autonomous Underwater Vehicle, which was controlled using OS-9.

Programming with Solder

This new column will appear from time to time as material is available. Primarily, I expect, it will consist of hardware projects for the Color Computer, though submissions for other hardware that uses OS-9 are very welcome. Skill level required for assembling these projects will vary with the project.

Submissions and Upcoming Articles

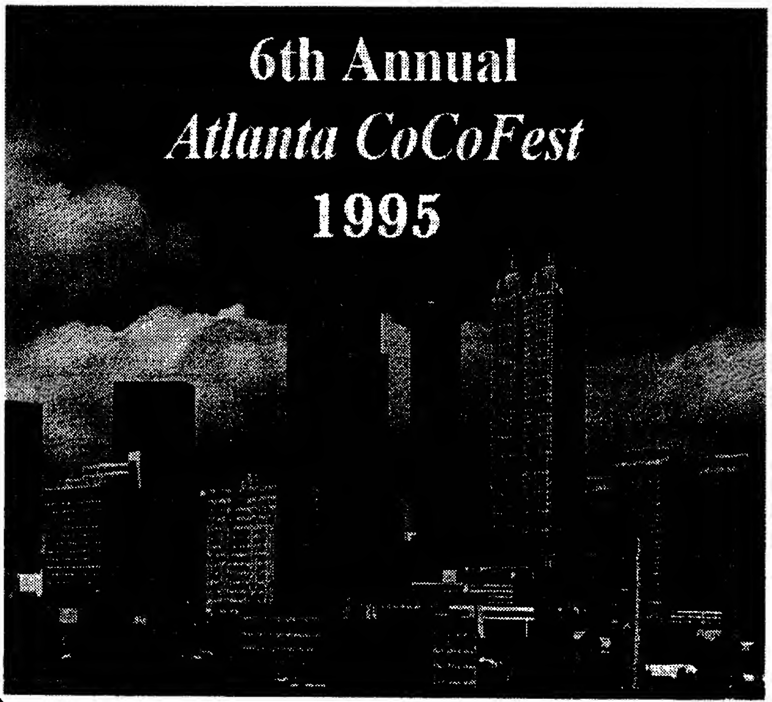
Submissions, as always, are welcome. Just send it by email or snail mail to the MOTD Editor at any of the Users Group addresses.

Colin McKay
President

6th Annual Atlanta CoCoFest 1995

Holiday Inn, Northlake

Tucker, GA (Atlanta area)



Show Hours:

```
Sat. Sep 30 9:00 AM - 5:00 PM
Sun. Oct 1  9:00 AM - 3:00 PM
```

Vendor Setup:

```
Fri. Sep 29 6:00 PM - 9:00 PM
Sat. Sep 30 8:00 - 8:45 AM
```

Admission:

$10.00 (Whole show)

Reservations:

Holiday Inn, Northlake
800-465-4329 or 404-938-1026

Sponsored by:

Atlanta Computer Society
PO Bos 80694
Atlanta, GA 30366
BBS: 404-636-2991

Minutes for Board of Directors Meeting July 5, 1995

A regular meeting of the OS-9 Users Group Board of Directors (BOD) was called to order at 7:51 PM CT on July 05, 1995 on Delphi. Present: Colin McKay, president and chairman of the Board, directors-at-large Ed Gresick, and Eddie Kuns. Present as invited guests: Howard Luckey, secretary, Mike Rowen, VP-6809, Joel Hegberg, editor of MOTD, and Boisy Pitre, member.

The first order of business was the approval of the minutes of the last regular meeting of the BOD held on 03 May 1995. Minutes approved as presented.

The second order of business was the treasurer's report. A report by Brother Jeremy showing deposits and withdrawals for the month of May was accepted and approved.

OLD BUSINESS:

Item 1. Vendor's package:
Colin has a form he is using and will send a copy to Ed for his comments.

Item 2. Joel Hegberg's proposed OS-9 Clipboard Standard:
Skipped for later discussion during the meeting.

Item 3. Letter of appreciation:
The President reported that a letter of appreciation had been sent to Kurt Johnson of KD Consulting for his seminar on Sculptor presented at the last Chicago Fest.

Item 4. Presidential spending:
This is a proposal to establish a Board policy that would limit the amount the president can spend without Board approval to $100.00. Presently are there no limits on spending other than the discretion of the President and the Treasurer. Resolution postponed.

Item 5. MOTD:
Latest issue should be in the mail soon.

Item 6. Library status:
Dave Kelly from OCN has agreed to serve as UG librarian. Arrangements were made to mail our library to him.

Item 7. Atlanta Fest:
The President reported that the Atlanta Computer Society has agreed to swap booth space for ads in the MOTD.

The BOD returned to the discussion of item 2 of old business, Joel's clipboard. One question is whether the UG through the BOD should endorse things or just announce that things are available. The issue is not the Clipboard itself but the related issue of standards and how much the UG should or should not set them. The BOD has asked Brian Goers, VP-OSK, to chair a committee to get things started.

NEW BUSINESS:

Item 1. OS-9 Scholarship:
Boisy Pitre has suggested that the UG offer a scholarship. After some discussion a committee made up of Boisy, chair, Ed Gresick, and Joel Hegberg was formed to work out final details for the scholarship. Some of the details include: tax status of the UG so that contributions can be tax deductible, qualifications to receive the scholarship, publicizing the program, etc. No formal Board action was taken.

The meeting adjourned at 8:55PM.

Howard Luckey, Secretary
Tuesday July 25, 1995

These minutes were approved as read at the September 6, 1995 meeting of the Board of Directors.

OS-9 WWW Home Page
`http://www.wisc.edu/~pruyne/os9faq.html`

Users Group WWW Home Page
`http://www.wisc.edu/~pruyne/os9ugfaq.html`

OS-9 Online Conference Schedules

DELPHI OS-9 Late Night Conferences with Chris Perrault

```
Monday, 18 Sep 95 - 10PM ET - Upcoming Atlanta Fest
Monday, 25 Sep 95 - 10PM ET - KD Consulting discusses Sculptor database
```

GENIE CoCo Real Time Conferences

```
Every Sunday - Open discussion forum  9-10PM ET
```

OS-9 IRC Conferences

```
Sunday, 10 Sep 95 - 10PM ET (GMT - 4) : OS-9 Users Group projects
Sunday, 24 Sep 95 - 10PM ET (GMT - 4) : Open Forum

Conferences are held every second Sunday at this same time.
Type  /join #os9  once inside IRC.
```

Are there any other on-line conferences happening out there? If you know of other conferences or events that are taking place on other on-line services (AOL, Genie, CIS, etc.) please send mail to me at either address listed below. I will post any information I receive:

Mike Rowen
VP OS-9/6809
OS-9 Users Group
mrowen01@delphi.com

OS-9 Periodical List

Compiled by Ed Jones

MAGAZINES	PUBLISHER	PRICE	ADDRESS
the world of 68' micros	Frank Swygert	$25.00 USA $32.00 Canada (8 issues; one year)	Farna Systems PO Box 321 Warner Robins, GA 31099-0321 (912)328-7859 (voice)
MicroDisk	Farna Systems	$40.00 USA $44.00 Canada	(See 68' Micros above)
OS-9 International	European Forum for OS-9	Swiss Francs for one year (three issues): Switzerland: 25.00 Europe: 30.00 Overseas: 35.00	EFFO, PO Box CH-8606 Greifensee, Switzerland (os9int@effo.ch)
Nine Times	JWT Enterprises	$34.95 USA $35.95 Canada (6 issues; one year)	JWT Enterprises 5755 Lockwood Blvd. Youngstown, OH, 44512 (216)758-7694 (voice)
MOTD	OS-9 Users Group Inc.	$25.00 USA and Canada $30.00 others	OS-9 Users Group, Inc. 6158 West 63rd Street Suite #109, Chicago, IL 60638
OS-9 Newsletter	Bellingham OS-9 UG	$10 (12 issues)	OS-9 Newsletter 3404 Illinois Ln Bellingham, WA 98226-4238
CoCo 1-2-3	Glenside CoCo Club	$15.00 USA	Glenside CoCo Club RR #2, Box 67 Forrest, IL 61741-9629 (708)428-3576 (voice) (708)428-0436 (BBS)

Straight From the Horse's Mouth

by Colin McKay

The following article contains highlights from the OS-9 DAVID (Digital Audio/Video Interactive Decoder) Conference held on Delphi, 31 July 1995. Permission has been granted by the participants for reproduction in the OS-9 Users Group newsletter, and posting to Internet newsgroups.

The Conference was hosted by David Kimble from Microware Systems Corp in Des Moines, Iowa. David was speaking for himself during this conference, not as an official Microware representative. Thanks also to Chris Perrault for arranging the conference. This text was heavily edited for brevity and clarity, so my apologies for any errors that may have been introduced.

The transcript includes comments from the following participants:

DKIMBLE:	David Kimble (special guest host)
Chris:	Chris Perrault (organizer)
JOELHEGBERG:	Joel Hegberg
Hugo:	Hugo Bueno
HAWKSOFT:	Chris Hawks
Mike_Rowen:	Mike Rowen
Colin:	Colin McKay
Boisy:	Boisy Pitre
jb:	John Baer
DOMM:	Domenick Mirando

This is not a complete list of attendees, of course; it only includes those who made comments captured in the transcript.

OS-9 DAVID Conference
Held on Delphi, 31 July 1995

Chris> David, please give us a bit of a summary of DAVID for those not familiar with it.

DKIMBLE> OK, I'll start with a little history as to how Microware came to be in the interactive TV game. When I'm done I'll take any questions.

As you may know, Sony and Philips came to us back in the mid-80's wanting to create this new generation of consumer electronics that they would call CD-i. They needed a real-time operating system that didn't need 4mb to run in and had a chance of running for some period of time without having to press CTL-ALT-DEL. :)

They approached Microware, and CD-RTOS was born. CD-RTOS is nothing but OS9 as everyone knows it, with a few new I/O subsystems thrown in. Primarily CD support, which introduced the concept of asynchronous I/O. (You do a read call, it returns immediately, you get signals telling you how the read is doing.)

The other I/O subsystem that was new was the MPEG full motion video support. This was all nice and cool and everyone was happy in Des Moines.

Then along came the Telcos with beaucoup bucks saying they want to do this thing called Interactive TV (ITV). Never mind what the hell it is, they just know they have to have it.

So we (Microware) thought, "hmm". This ITV is pretty close to this CD-i problem we already solved. In fact, if you pretend that at the other end of this long telephone cable, the phone company had nothing but a CD device, it was almost identical. So, thanks to our good advice to the phone companies, the first generation of ITV boxes looked one hell of a lot like CD-i players with a remote CD. They had 68k CPUS, they had CD-i graphics chips (MCD211 & 212), and they had MPEG chips similar to the ones in the CD-i video cartridges.

The three main problems to solve were:

1) the MPEG chips chosen were slightly different, requiring some driver work;
2) the paradigm of a CD had to be abstracted to a network; and
3) no one had done this before (or understood all that was involved).

DKIMBLE> So, that first generation worked out pretty well... IBM had a division in NY that made set-top boxes, and another company, Adaptive Microware (no relation) made them as well. They were what came to be called ADSL boxes. ADSL is approximately Asymetrical Digital Subscriber Line. It allowed (as long as you had fiber within 18000 feet of your home) for the T1 video pipe (1.5MB downstream - notice the data rate is 1xCD), the X.25 control channel (two-way 16kb), and the POTS (plain old telephone service) to all be multiplexed together on standard phone wires! This was fine, and actually worked pretty well on the set-top. Of course, the network was a little more complicated than that and things developed slower than anticipated.

None of what the telcos were trying to do had been done before. The old pioneers getting the arrows in the back applies here. They were also hampered by the fact that regulation in these good ole United States is less than favorable if you're a phone company. Particularly when this started (1982-83), there was some fear about what the government types would do when the phone companies started delivering video.

Well, there are now trials going on across the world: Hong Kong, Japan, Italy, and the UK, with more planned: France, Singapore, Sweden, Germany, Canada, and more.

The ADSL model is losing favor, in favor of pure digital networks (ATM primarily). The next generation of boxes have some other nice things as well: MPEG2 decoders, PPC CPU's, more interesting gfx chips.

Our work continues to be that of helping the network operators determine what they really need (vs. what they sometimes want), and helping set-top manufacturers port all of the I/O drivers to their boxes, and making sure that the server people all talk a language that all of the boxes can understand (eg. DSM-CC).

There are a lot of cool things planned for this technology, but there are so many components that the end-to-end network is taking longer to come up than originally planned.

The OS works, however :), and we are trying our best to bring the rest of the players up to where we can turn on a real network. Toms River, NJ looks to be the first real end-to-end interactive network - due to come online in less than a month. It will have Philips STBs, a BroadBand Tech (BBT) network, and OS9/DAVID running in the boxes.

JOELHEGBERG> (STB = set top box)

DKIMBLE> DAVID is to ITV what CD-RTOS is (was) to CD-i. A particular combination of I/O subsystems required to do ITV.

HAWKSOFT> Was?

DKIMBLE> Not really was... that was a side-ways reference to my disappointment at the popularity of CD-i...

DKIMBLE> Any questions?

Chris> Is the MPFM which I read about in an issue of Pipelines a couple of years ago part of the ITV and CD-I systems?

DKIMBLE> yes, the original MPFM (MPEG file manager) was designed to be compatible with the original spec of Philips. The most recent incarnation deviates for the first time from the "green book" (the CD-i spec). There were a number of things about the original implementation that were not, how shall I say, the best approach.

Hugo> how does HDTV fit into all this?

DKIMBLE> Well, from OS9's point of view, it's just another set of h/w to write drivers for (encoders, gfx chips, etc.). It will be important to us when a network provider specifies that they want HDTV in their network. It will require higher bandwidth networks (30 frames/sec at much higher resolution), but those are coming now.

Mike_Rowen> Do you envision cable providers taking the first step instead of telcos?

DKIMBLE> Well, cable providers are not as far along as telcos are and they are not as rich. They also suffer from customer perception that they have been less than.... I don't know the word, but that they have lousy service :)

Mike_Rowen> :)

HAWKSOFT> 500 channels and nuthin on!

DKIMBLE> There are a large number of folks that are ready for the telcos (come home to ma bell) to take over their ITV.

Mike_Rowen> Will the new CD format rivalry impact things for Microware?

DKIMBLE> The new CD format rivalry means extra drivers, that's all...

Mike_Rowen> That's good to hear.

DKIMBLE> The beauty of where we sit is that everything is Legos (tm) i.e. a new h/w component has very little influence on the rest of the system (data rates are really the biggest impact).

Colin> When can the consumer expect to actually start seeing these things, both in the US and in Canada?

DKIMBLE> When can we expect to start seeing things? Good question. There are currently 20 companies building DAVID boxes. There are all sorts of networks being built. It will be a number of years (4-5) until these things are all over.

Colin> Thanks. Any idea of the cost?

DKIMBLE> Hah! Depends on who you ask. There are those that say they want a PPC CPU, 8MB RAM, a $250 network interface module (NIM), and a toaster on the side...

Chris> How much for the toaster? <G>

DKIMBLE> Then there are those that want it to be <$300. Those two groups never seem to go to lunch together. :)

JOELHEGBERG> :)

Chris> Will the end user have to change anything in his hardware, ie: Better television upgrades etc?

DKIMBLE> About the h/w, we are trying very hard to convince manufacturers to build things so that they will be more network independent. i.e. the first thing you do on a network is download the protocol stack for that network. In terms of TV resolution, the current boxes all put out NTSC (normal North American TV) resolution, so there is no special requirement.

Mike_Rowen> What kinds of interactive services are being offered and from what hosts?

DKIMBLE> The current idea is that MMDS (lower-end) services will be the first real implementation, and those boxes can actually be built for $300.

Hugo> MMDS?

DKIMBLE> Multi-channel, Multi-point, D<something>, S<something>. It's a wireless network, using line-of-sight microwave antennae/receivers. The current back channel will be a modem, kind of like the RCA stuff does now

Hugo> What exactly does ITV provide?

DKIMBLE> What can you do <big voice>: "I'm glad you asked that question, son."...

JOELHEGBERG> ;)

DKIMBLE> There are plans to do interactive shopping, Video on demand (movies when you want them), networked games, let's see..., on-line banking... Electric Program Guides (while you're watching TV, you have access to TV-guide like info on line).

JOELHEGBERG> Don't forget Karaoke, Dave!

DKIMBLE> Oh yes, karaoke! Microware is currently developing an algorithm to randomly play a video with karaoke audio and guarantee that they have nothing in common...:)

jb> heh!

Boisy> I know, I've seen it. :)

Colin> Oh, you mean a Microsoft 1.0 Release.

JOELHEGBERG> ooooh.. :)

Chris> LOL

DKIMBLE> There are people working on Interactive TV, as in play along with Jeopardy.

Hugo> Can personal computers be hooked up to the "wire"?

DKIMBLE> Personal computers can certainly, as long as they're running OS9 <G>...

JOELHEGBERG> Make sure that's in the log.

HAWKSOFT> YEAH!!!

Chris> Yeah, can we hit the homepages with our tv's?

DKIMBLE> There are people right now putting a DAVID box on a PC board. There are people right now also writing HTML interpreters to netsurf from your couch... (god help us reading text on a TV). I saw one presentation where they had mocked up a Jeopardy game where you played along with the real contestants. If you get a good enough score at the end of the game, the sponsor sends you a nice package.

Hugo> I just hope the content won't be more of the same fluff we get now.

DKIMBLE> I am not all that encouraged in this regard...

Hugo> Another bothersome thought is that perhaps the providers will be able to monitor people's watching habits.

DKIMBLE> The people who will decide what goes on these networks will be the ones who are spending and making the $. Yeah, there is an element of paranoia that is probably healthy.

Chris> Do you expect a lot of exposure once a succesful network is established in Toms River (the test site selected by Bell Atlantic)?

DKIMBLE> Toms River will certainly create some headlines. It remains to be seen what the "spin" will be.

DOMM> How long is the Toms River experiment supposed to last?

DKIMBLE> Not sure about that. If it goes bad, probably not long <G>.

Chris> Is 'Modular Windows' completely out of the picture or is Microsoft still sticking its foot in the door?

Hugo> There must be competition out there. RCA DSS is one, what are the others?

DKIMBLE> Microsoft is still in the game -- when something is that big, everyone keeps an eye on it. They are proposing an end-to-end Microsoft solution that has some people getting deja-vu. RCA DSS is the big one, they're pushing the envelope.

jb> DSS is nice, but not interactive <g>.

JOELHEGBERG> I think Microware's solution is the most mature in the market, however.

Chris> and it sounds like a natural fit

Mike_Rowen> I work for EDS and they are interested in the supply side of things. Have you worked with them?

DKIMBLE> We have talked with all sorts of people. There will be a place for the types of services that EDS has to offer. Right about DSS, and that's the thing - it has flash, people are impressed to see digital video for the first time. It's just MPEG1 video being delivered by satellite. As Joel said, it's really not as mature as what currently exists using OS9, but it's taking longer to get the more complicated networks on line.

jb> Yep! Got to build those.

DOMM> If one had to choose now between hooking up with cable or RCA...???

Hugo> Domm, I would choose neither :-)

DKIMBLE> DSS is not cheap. Cable has the physical plant too, giving them an edge price-wise.

DKIMBLE> I need to wrap up soon. Any other questions?

Hugo> Thanks for stopping by David, it was interesting!

JOELHEGBERG> Anyone with one more question for Dave?

Mike_Rowen> Great conference. Appreciate your time David.

jb> I'll second that!

DKIMBLE> Glad to preach the gospel of OS9 <G>

JOELHEGBERG> Yeah, this has been great...

Chris> Thanks for your insights Dave. This topic definately lived up to it's expectations!

DKIMBLE> it's fun work, I like talking about it.

JOELHEGBERG> Glad you set it up, Chris, and glad you could come, Dave!

DKIMBLE> my pleasure!

The Toolbox

A long listing this month, courtesy of Bob van der Poel. HPSET can be used to set up HP-compatible printers (both laser and deskjet). A number of interesting techniques are used, including Termcap and an interesting technique of an array of structures with function pointers.

HPSET
To use the program, just type "hpset <enter>". You will be presented with a screen of options and a cursor bar. Move the cursor bar with the <j> and <k> keys. To change an options, use <h> and <l>. Once you have the options set to your liking, press <ENTER> and the printer will be setup. If you press <q> the program will exit without sending the data to the printer.

By default, the printer name is "/p". You can change this by using the -p option when start hpset. For example:

```
hpset -p=/pl
```

Of course, since this distribution includes the source you could change it there...

To compile the program, use the following command line:

```
cc hpset -i -l=/dd/lib/termlib.l
```

Some of the various options will not have any effect on the printer... depending on what fonts, etc. you have available. For example, on my printer "light font weight" has no effect; I don't have "italic bold"; etc. Of course, if you have more builtin fonts or have downloaded soft fonts to the printer this may well work.

Bob van der Poel Software
PO Box 355, Porthill, ID, USA 83853
PO Box 57, Wynndel, BC, Canada, V0B 2N0
Email: CIS 76510,2203

```
/* hpset.c */

#define MAIN

#include <stdio.h>
#include <termcap.h>
#include <sgstat.h>

/* These function declarations are needed to initialize the cmds[] structure.
*/

char *show_reset(), *show_orient(), *show_fsize(),
     *show_fstyle(), *show_value(), *show_wrap(),
     *show_eol(), *show_lmargin(), *show_weight();

void out_reset(), out_fsize(), out_fstyle(), out_value(), out_weight();

/* This structure contains everything we need to know about each
   printer function. Note that we have included '%d's in the
   codes...these are replaced by actual values when they are
   sent to the printer. Note that the order may be important...for example,
   you most probably should make sure that RESET is the first function!
   Other functions may depend on order (eg. margins may depend on font).
*/

struct
{
     char *name;         /* option name */
     int  default,       /* default value */
          min,           /* minimum value */
          max,           /* maximum value */
          new;           /* new value set by user */
     char *codes;        /* printer codes for option */
     char *(*dfunc)();   /* function returning option setting name */
     void (*outfunc)();  /* function to send codes to printer */
}cmds[]=
{
     "RESET",             1, 0,   1, 0, "\x1bE",      show_reset,   out_reset,
     "Orientation",       0, 0,   1, 0, "\x1b&l%dO",  show_orient,  out_value,
     "End of Line Wrap",  1, 0,   1, 0, "\x1b&s%dC",  show_wrap,    out_value,
     "Line Termination",  0, 0,   3, 0, "\x1b&k%dG",  show_eol,     out_value,
     "Font Size",         0, 0,   2, 0, "\x1b(s%dH",  show_fsize,   out_fsize,
     "Font Style",        0, 0,   1, 0, "\x1b(s%dS",  show_fstyle,  out_value,
     "Font weight",       1, 0,   2, 0, "\x1b(s%dB",  show_weight,  out_weight,
     "Copies",            1, 1,  99, 0, "\x1b&l%dx",  show_value,   out_value,
     "Lines per Inch",    6, 1,  48, 0, "\x1b&l%dD",  show_value,   out_value,
     "Page Length",      60, 1, 200, 0, "\x1b&l%dP",  show_value,   out_value,
     "Left Margin",       0, 0,  40, 0, "\x1b&a%dL",  show_lmargin, out_value
};

FILE *outpath;
char *printer="/p";

int  nrows,       /* number of rows on screen */
     ncolumns,    /* number of columns on screen */
     numcmds,
     termcap_ok;

extern int errno;

#define TCAPSLEN 1024

/* Termcap variables */

char PC_,        /* pad character */
     *BC,        /* cursor backspace */
     *UP;        /* cursor up */

short ospeed;    /* terminal speed */

static char *CM,     /* cursor movement */
            *CL,     /* clear screen    */
            *SO,     /* standout start  */
            *SE;     /* standout end    */
```

```
/* ====================================================================
   Inititialize the termcap stuff
*/

init_termcap()
{
     register char *term_type;
     register int t;
     static char tcapbuf[TCAPSLEN];
     static char tcbuf[1024];
     char *ptr;
     char *temp;

     extern char *getenv();

     static struct
     {
          char *tcname;
          char **tcptr;
     }tc[]=
     {
          "cm", &CM,
          "cl", &CL,
          "so", &SO,
          "se", &SE
     };

     if((term_type=getenv("TERM"))==0)
          terminate("Environment variable TERM not defined!");
     if(tgetent(tcbuf, term_type)<=0)
          terminate("Unknown terminal type '%s'!",term_type);

     ptr=tcapbuf;

     if(temp=tgetstr("pc", &ptr)) PC_=*temp;

     for(t=0; t<sizeof(tc)/sizeof(tc[0]); t++)
     {
          if((*tc[t].tcptr=tgetstr(tc[t].tcname, &ptr))==NULL)
               terminate("Termcap needs '%s' entry\n", tc[t].tcname);
          if(ptr>=&tcbuf[TCAPSLEN])
               terminate("Termcap description too big");
     }

     nrows=tgetnum("li");
     ncolumns=tgetnum("co");

     if(!nrows || !ncolumns)
          terminate("Unable to determine screen size");
     if(ncolumns<80 || nrows<24)
          terminate("Screen must be at least 80x24");
     termcap_ok=1;
}

putpad(s)
char *s;
{
     extern int write1();  /* function to print 1 character */
```

```
     if(s) tputs(s,1,write1);
}

write1(c)
char c;
{
     putc(c,stdout);
}

gotoxy(x,y)      /* position cursor at specified x/y */
int x,y;
{
     putpad(tgoto(CM,x,y));
}

clrscrn()      /* Clear video screen */
{
     putpad(CL);
}

revon()      /* Turn on reverse video */
{
     putpad(SO);
}

revoff()      /* Turn off reverse video */
{
     putpad(SE);
}


/* ================================================
   Set up the terminal
*/

static struct sgbuf pathbf1, pathbf2;
static char pathbfv=0;     /* flag, 1==path ok to reset */


setterm1()
{
     setbuf(stdin, NULL);     /* single char input */

     if(!pathbfv)
     {
          _gs_opt(1,&pathbf1);
          _gs_opt(1,&pathbf2);
          pathbfv++;               /* signal okay to restore */
          pathbf1.sg_echo=
          pathbf1.sg_pause=
          pathbf1.sg_eofch=
          pathbf1.sg_psch=
          pathbf1.sg_kbich=
          pathbf1.sg_kbach=0;
     }
     _ss_opt(1,&pathbf1);
     ospeed=pathbf2.sg_baud;      /* set terminal speed for termcap */
}
```

```
resterml()
{
     if(pathbfv) _ss_opt(1, &pathbf2);
     fflush(stdout);
}


/* ===================================================================
   Display usage message
*/

help()
{
     register int t;
     static char *msg[]=
     {
          "Syntax: HPSet <opts>",
          "Options:",
          "\t-p=path     specify printer",
          "\t-?          this message"
     };
     for(t=0; t<sizeof(msg)/sizeof(msg[0]); puts(msg[t++]));
}


/* ========================================================
   Disaster exit
*/

terminate(s, p1, p2, p3, p4, p5)
char *s, *p1, *p2, *p3, *p4, *p5;
{
     if(termcap_ok)
     {
          gotoxy(ncolumns-1, nrows-1);
          putc('\n', stdout);
     }
     if(s)
     {
          printf(s, p1, p2, p3, p4, p5);
          putc('\n', stdout);
     }
     resterml();
     exit(0);
}


/* ===================================================================
   This set of functions are used by showline() to determine the
   'value' to print for a options.
*/

char *
show_reset(value)/* reset printer yes/no */
int value;
{
     static char *msg[]=
     {
          "No",
          "Yes"
     };
     return msg[value];
}

char *
show_orient(value)     /* page orientation */
int value;
{
     static char *msg[]=
     {
          "Portrait",
          "Landscape"
     };
     return msg[value];
}

char *
show_fsize(value)     /* font pitch (cpi) */
int value;
{
     static char *msg[]=
     {
          "Pica (10cpi)",
          "Elite (12cpi)",
          "Compressed (16.6cpi)"
     };
     return msg[value];
}

char *
show_fstyle(value)     /* typestyle (roman/italic) */
int value;
{
     static char *msg[]=
     {
          "Roman (upright)",
          "Italic"
     };
     return msg[value];
}

char *
show_wrap(value)     /* EOF wrap on/off */
int value;
{
     static char *msg[]=
     {
          "Enabled",
          "Disabled"
     };
     return msg[value];
}

char *
show_eol(value)     /* EOL handling */
```

```
int value;
{
     static char *msg[]=
     {
          "CR=CR, LF=LF, FF=FF",
          "CR=CR+LF, LF=LF, FF=FF",
          "CR=CR, LF=CR+LF, FF=CR+FF",
          "CR=CR+LF, LF=CR+LF, FF=CR+FF"
     };
     return msg[value];
}

char *
show_lmargin(value)     /* Left margin handling */
int value;
{
     static char buff[20];
     sprintf(buff, "%d characters", value);
     return buff;
}

char *
show_weight(value)     /* Typeface weight */
int value;
{
     static char *msg[]=
     {
          "Light",
          "Medium",
          "Bold"
     };
     return msg[value];
}

char *
show_value(value)     /* numeric values */
int value;
{
     static char buff[20];
     sprintf(buff, "%d", value);
     return buff;
}

/* ==================================================================
   These functions send the option value to the printer. They are needed
   since some cmds[] values need to be converted to printer values.
*/

void
out_reset(n)     /* reset...don't send if value==0 */
register int n;
{
     if(cmds[n].new) fprintf(outpath, cmds[n].codes);
}

void
out_fsize(n)     /* font size (pitch) */
register int n;
{
     register int t;
     switch(cmds[n].new)
     {
          default:
          case 0: t=10; break;
          case 1: t=12; break;
          case 2: t=17; break;
     }
     fprintf(outpath, cmds[n].codes, t);
}

void
out_weight(n)
{
     register int t;
     switch(cmds[n].new)
     {
          default:
          case 0: t=-3; break;
          case 1: t=0; break;
          case 2: t=3; break;
     }
     fprintf(outpath, cmds[n].codes, t);
}

void
out_value(n)     /* send a numeric value */
register int n;
{
     fprintf(outpath, cmds[n].codes, cmds[n].new);
}

/* ==================================================================
   Display an option line on the screen
*/

showline(n, mode)
int n, mode;
{
     register char *p1, *p2;
     register int l1, l2;

     gotoxy(5, n+2);
     if(mode) revon();

     p1=cmds[n].name;
     p2=(*cmds[n].dfunc)(cmds[n].new);

     l1=strlen(p1);
     l2=strlen(p2);

     fputs(p1, stdout);
     while(l1++<20) putc(' ', stdout);
     fputs(p2, stdout);
     while(l2++<30) putc(' ', stdout);

     if(mode) revoff();
}


/* =================================================================
   M A I N
*/

main(argc, argv)
int argc;
char *argv[];
{
     register int t, key, n;
     register char *p;

     for(t=1; t<argc; t++)
     {
          p=argv[t];
          if(*p++!='-') usage();
          switch(tolower(*p++))
          {
               case ‘p’:
                    if(*p=='=') p++;
                    printer=p;
                    break;

               default:
                    usage();
          }
     }
     init_termcap();
     setterml();

     for(t=0; t<sizeof(cmds)/sizeof(cmds[0]); t++)
          cmds[t].new=cmds[t].default;

     clrscrn();
     gotoxy(0,0);
     printf(“      HPSET - HP Printer Setup -- (c) 1995, Bob van der Poel Soft-
ware”);
     gotoxy(0,nrows-1);
    printf(“KEYS: j/k move cursor, h/l change setting, q to quit, <ENTER> sets
printer”);

     for(t=0; t<sizeof(cmds)/sizeof(cmds[0]); t++) showline(t, 0);

     for(t=0; ; )
     {
          showline(t, 1);
          key=getc(stdin);
          showline(t, 0);

          switch(tolower(key))
          {
               case ‘j’:
                    if(t< sizeof(cmds)/sizeof(cmds[0])-1) t++;
                    else t=0;
                    break;
               case ‘k’:
                    if(t) t--;
                    else t=sizeof(cmds)/sizeof(cmds[0])-1;
                    break;
               case ‘l’:
                    if(cmds[t].new<cmds[t].max) cmds[t].new++;
                    else cmds[t].new=cmds[t].min;
                    break;
               case ‘h’:
                    if(cmds[t].new>cmds[t].min) cmds[t].new--;
                    else cmds[t].new=cmds[t].max;
                    break;
               case ‘\n’:
                    if((outpath=fopen(printer, “w”))==NULL)
                         terminate(“Can’t open path to ‘%s’, error %d”,
                         printer, errno);
                    for(n=0; n<sizeof(cmds)/sizeof(cmds[0]); n++)
                         (*cmds[n].outfunc)(n);
                    fclose(outpath);
                    terminate(“Printer codes sent”);
               case ‘q’:
                    terminate(“\x07*** PRINTER NOT SET ***”);
          }
     }
}

usage()
{
     static char *msg[]=
     {
          “HPSET (c) Bob van der Poel Software, 1995”,
          “Syntax: Hpset [opts]”,
          “Usage: Sets options on HP Laser Printer”,
          “Options:”,
          “    -p[=]pathset printer port (default=/p)”
     };
     register int t;

     for(t=0; t<sizeof(msg)/sizeof(msg[0]); puts(msg[t++]) );
     terminate(NULL);
}
```

Vendor Listing

Vendor	Address	Phone
AniMajik Productions	4650 Cahuenga Blvd, Ste #7, TOLUCA LAKE CA 91602	(818)-769-1938
Ark Systems USA	PO Box 23, SANTA CLARA CA 95052	(408)-244-5358
* Blackhawk Enterprises	PO Box 10552, ENID OK 73706-0552	(405)-234-2347
* Bob van der Poel Software	PO Box 355, PORTHILL ID 83853-0355	(604)-866-5772
Computer Design Lab	RR1 Box 36, RHINELAND MO 65069	(314)-236-4373
Computer Design Services	2550 Sandy Plains Rd, MARIETTA GA 30066	(404)-973-2170
CoNect	449 South 90th St, MILWAUKEE WI 53214	(414)-258-2989
* Delmar Co	PO Box 78, 5238 Summit Bridge Rd, MIDDLETON DE 19709	(302)-378-2555
Disto	1710 Depatie, ST LAURENT QC H4L 4A8, Canada	(514)-744-7777
Farna Systems	PO Box 321, WARNER ROBINS GA 31099-0321	(912)-328-7859
Frank Hogg Laboratory	204 Windemere Rd, SYRACUSE NY 13205	(315)-469-7364
Gale Force Enterprises	P.O. Box 631, Surrey Main, SURREY BC V3T 5L9, Canada	
* Hawksoft	244 S Randall Rd, ELGIN IL 60123	(708)-742-3084
The KD Consulting Group, Inc.	5768 Fairridge Lane HAMILTON OH 45011	(800)-837-1619
KEN-TON	Contact Farna Systems	
Microware Systems Corporation	1900 NW 114th St, DES MOINES IA 50322	(515)-224-1929
* Northern Xposure	7 Greenboro Cres, OTTAWA ON K1T 1W6, Canada	(613)-736-0329
Peripheral Technologies	1480 Terrell Mill Rd #870, MARIETTA GA 30067	(404)-973-2156
RemComs	100 Whitsunday Dr, KIRWAN QLD 4817, Australia	+61 77 734884
Sub-Etha Software	PO Box 152442, LUFKIN TX 75915	
Windsor Consulting Group	2014 Cherokee Pkwy, Suite J, LOUISVILLE KY 40204	(502)-454-0054
* Wittman Computer Products	7376 North Bergen Rd, BERGEN NY 14416	(716)-494-1506

* Indicates corporate member of the OS-9 Users Group

A Basic09 Tutorial

by Bob Devries

This article is a continuation of the Basic09 tutorial by Bob Devries from Australia. One or two articles from the series will appear each issue, depending on space. (Real Queen's English spelling mode enabled...)

Part 3 of 14

Listing 1 is the BASIC code for the numbersquare programme from 'Microcomputing', June 1981. It is written in vanilla BASIC, but is suitable for Extended Colour Basic with minor modifications, in lines 140, 490, 760, 920, 960 and 1450. Can you work out what to change? (Note please that a colon is used instead of a REM)

The game is a fairly simple one based on the use of multidimensioned arrays. Note the use of colons at the beginning of REM lines, which is also possible in DECB. Now comes the tricky part, the conversion to Basic09. Firstly I'll show you the code as I rewrote it (Listing 2), then I will explain it.

OK, so here we go. First, I used the command BASE 1. This is because all the arrays in the BASIC programme start at 1, and Basic09 usually starts at zero (actually, BASIC does too, but I see no reason to waste valuable memory). Next, I dimensioned all the variables and arrays, including a variable type which you may not have seen before, the BOOLEAN type. This variable may only contain either TRUE or FALSE! I used the SHELL command to turn off the OS9 screen pause, so that the programme won't sit there waiting at what it thinks is the end of a screen. That can get a bit confusing!

The next thing you must realise is that I have used NO LINE NUMBERS! This is really the best way to programme. Sure, Basic09 will allow their use, but the code is much more elegant without them, if a little more difficult to write. I set up a loop to allow the choice of whether to play another game which is done in line 1450 in the BASIC version. I used a WHILE loop here so that all I need to do is make a variable FALSE to exit the loop.

Next, after clearing the screen (printing a formfeed character), I print the opening message and ask the player for his choice of game. This follows through to line 160 of the BASIC programme. On the basis of the player's answer I RUN either the procedure 'setupssb' or 'setupmsb'. You may notice a slight variation here, I only tested for a '1' to select sequential, and any other key would run the magic square option. Then the next 38 lines do the same as lines 480 to 700 in the BASIC programme. You will notice that the BASIC programme uses quite a large number of GOTOs in this piece of code to break out of various parts of the code to continue the FOR-NEXT loop. I simply set a variable (lop) to TRUE and again used a WHILE loop. All the other variables have the same names, although you can of course use any name, and are not limited to two significant characters as in BASIC.

To make the screen easier to display, I used in this case a gfx2 command 'cwarea'. This command limits the area of the screen to be printed to, and means I don't need to use cursor maneuvering code at every print line. I simply resized the screen to a 22 by 12 character box in approximately the middle of the (80 by 24) screen. So next I print the scrambled array to the screen with one space so that pieces may be moved around. The game is played by entering the number of the square which you want to move into the space. The programme checks to see if the number is one of the ones adjoining the space. A tone is sounded, and you are prompted for input. Pretty standard here, 'though I could just have used 'PRINT CHR$(7)', but the gfx2 'bell' command is nicer. If the player enters zero, the programme resets the screen back to its 80 by 24 size, and clears the screen and quits.

The rest of the code is fairly straight forward, again the original uses many GOTOs to quit out of FOR-NEXT loops (not a good practice in my opinion, even in BASIC), and I have used boolean tests for this purpose. For every move made, the programme checks the array against the solution, and if the last move solves the puzzle, it prints the necessary result.

One of the hardest parts of the conversion is keeping track of the variables, and the various loops. Basic09 is a bit unforgiving about 'UNMATCHED CONTROL STRUCTURES' so you can't stop doing a conversion such as this in the middle, without generating a series of error messages when you quit the editor. One way around this is to use a text editor (like VED or SCRED or SLED) to create the source code first, and then to load it into Basic09. The only thing you MUST do in this case is to make the word 'PROCEDURE' in UPPERCASE the FIRST WORD in the file. The letter P of procedure must be the first character in the file, or Basic09 will not recognize it.

OK, so there you have it. I would love to hear from you regarding your own trials and tribulations with Basic09 programmes, even if you don't really want to start out on conversions of this type, but are having difficulties managing some aspect of Basic09. Please write to me care of the newsletter editor, and let 'Professor Bob' help sharpen your programming skills.

Regards,Bob Devries
(bob@paxnet.com.au)

Wittman Computer Products

We have moved to a new location!!!

Wittman Computer Products
7376 North Bergen Road
Bergen, NY 14416

Same phone number...

Wittman Computer Products is proud to announce our upcoming release of a new OS-9 based computer. Running OS-9 V3.0, MGR a graphical environment interface, and utilize PC cards with its built in slots.

The product will be known as a WCP306 computer. It is based on a Motorola 68306 chip, specifications below.

Our target date to accept and fullfil orders will be the middle to end of June 1995. We are accepting orders now, *with money down.* The pricing of the machine will be:

WCP306 board only which includes OS-9 v3.0 & MGR $400.00 for the base package.

Single Board Computer with 16 bit PCAT I/O Bus
MC68306 CPU at 16.67 MHz , code compatible with 68000

Need more information, write or call us

Now available for the WCP306 or any OSK system, M6809 (Color Computer Emulator) by Bob Santy.

Purchase VED/VPrint with a WCP306 and receive a package price discount on VED/VPrint.

Look for our new logo on our products
You will find quality inside...

WCP

Ad was printed in this format to catch your attention

Wittman Computer Products

HARDWARE * SOFTWARE * CONSULTING

Full MM/1 and OS-9/68xxx support.

We offer Gift Certificates

Software

CirCad - Circuitry CAD program with EPS output
DeskTamer v2.0 - a premier personal information manager (calculator, filer, notetaker, full graphical interface)
InfoXpress for OSK & the Color Computer
LaTerm/LaDial - terminal package
Gold Runner 2000 - KWindows Game

Hardware

WCP306 NEW OSK machine
68340 Accelerator Boards
New 8 meg backplanes & I/O Boards for the MM/1
I/O Paddle Boards / Midi Boards
Floppy & Hard Drives
Cases & Power Supplies
Data / Fax Modems
Keyboards, Memory & Mice
Point-Of-Sale Equipment

William L. Wittman, Jr.
Wittman Computer Products
7376 North Bergen Road
Bergen, NY 14416
(716) 494 - 1506

our E-mail addresses are:
swamps.roc.ny.us!mm1wny!wlwjr
ww2150@acspr1.acs.brockport.edu

A Basic09 Tutorial

by Bob Devries

LISTING 1

```
0010 : Number square game
0020 : ver 4.0 - 12 nov 79
0030 : Marc I. Leavey, M.D.
0040 LINE= 0
0050 DIGITS= 0
0060 PRINT "N U M B E R   S Q U A R E S"
0070 PRINT "---------------------------"
0080 PRINT
0090 PRINT "WELCOME TO THE WORLD OF"
0100 PRINT "CONFUSION.  THERE ARE TWO"
0110 PRINT "VERSIONS OF NUMBER SQUARES:"
0120 PRINT " 1 - SEQUENTIAL"
0130 PRINT " 2 - MAGIC SQUARE"
0140 INPUT "WHICH IS YOUR PLEASURE",T
0150 IF T=1 GOTO 310
0160 IF T<>2 GOTO 140
0170 :
0180 : SET UP MAGIC
0190 : SQUARE BOARD
0200 :
0210 FOR I=1 TO 4
0220 FOR J=1 TO 4
0230 READ M(I,J)
0240 LET B(I,J)=M(I,J)
0250 NEXT J
0260 NEXT I
0270 DATA 1,6,15,8,12,11,2,5,10,13,4,3,7,16,9,14
0280 LET I1=4
0290 LET J1=2
0300 GOTO 440
0310 :
0320 : SET UP SEQUENTIAL
0330 : BOARD
0340 :
0350 DIM B(4,4)
0360 FOR I=1 TO 4
0370 FOR J=1 TO 4
0380 LET B(I,J)=(I-1)*4+J
0390 NEXT J
0400 NEXT I
0410 LET I1=4
0420 LET J1=4
0430 :
0440 : NOW SCRAMBLE THE BOARD
0450 : TWO HUNDRED TIMES
0460 :
0470 PRINT "I AM NOW SCRAMBLING THE BOARD..."
0480 FOR Q=1 TO 200
0490 LET M=INT(1+RND*4)
0500 ON M GOTO 510,560,610,660
0510 IF I1=1 GOTO 490
0520 LET B(I1,J1)=B(I1-1,J1)
0530 LET B(I1-1,J1)=16
0540 LET I1=I1-1
0550 GOTO 700
0560 IF I1=4 GOTO 490
0570 LET B(I1,J1)=B(I1+1,J1)
0580 LET B(I1+1,J1)=16
0590 LET I1=I1+1
0600 GOTO 700
0610 IF J1=1 GOTO 490
0620 LET B(I1,J1)=B(I1,J1-1)
0630 LET B(I1,J1-1)=16
0640 LET J1=J1-1
0650 GOTO 700
0660 IF J1=4 GOTO 490
0670 LET B(I1,J1)=B(I1,J1+1)
0680 LET B(I1,J1+1)=16
0690 LET J1=J1+1
0700 NEXT Q
0710 :
0720 : PRINT BOARD
0730 :
0740 LET M9=0
0750 : OUTPUT A "HOME UP"
0760 PRINT CHR$(16);
0770 PRINT "---------------------"
0780 FOR I=1 TO 4
0790 FOR J=1 TO 4
0800 PRINT": ";
0810 IF B(I,J)=16 PRINT "    ";:GOTO 840
0820 IF B(I,J)<10 PRINT " ";
0830 PRINT B(I,J);
0840 NEXT J
0850 PRINT ":"
0860 PRINT "---------------------"
0870 NEXT I
0880 :
0890 : ERASE REST OF SCREEN AND
0900 : BEEP FOR INPUT
0910 :
0920 PRINT CHR$(22);CHR$(7);CHR$(7);
0930 :
0940 : INPUT MOVE
0950 :
0960 INPUT "MOVE WHICH PIECE",M
0970 LET I1=0:J1=0
0980 FOR I=1 TO 4
0990 FOR J=1 TO 4
1000 IF B(I,J)=M THEN I1=I:J1=J
1010 NEXT J
1020 NEXT I
1030 IF I1=0 THEN PRINT "I CAN'T FIND THAT NUMBER":GOTO
940
1040 LET I2=0:J2=0
1050 FOR I=I1-1 TO I1+1
1060 IF I>4 GOTO 1090
1070 IF I<1 GOTO 1090
1080 IF B(I,J1)=16 THEN I2=I:J2=J1:GOTO 1170
1090 NEXT I
1100 FOR J=J1-1 TO J1+1
1110 IF J>4 GOTO 1140
1120 IF J<1 GOTO 1140
1130 IF B(I1,J)=16 THEN I2=I1:J2=J:GOTO 1170
1140 NEXT J
1150 LET M9=M9+1
1160 PRINT "NOT A VALID MOVE":GOTO 940
1170 LET B(I2,J2)=M
1180 LET B(I1,J1)=16
1190 ON T GOTO 1210,1320
1200 :
1210 : SEQUENTIAL SOLUTION
1220 :
1230 LET C=0
1240 FOR I=1 TO 4
1250 FOR J=1 TO 4
1260 IF B(I,J)<C GOTO 720
1270 LET C=B(I,J)
1280 NEXT J
1290 NEXT I
1300 PRINT "YOU GOT IT!"
1310 GOTO 1450
1320 :
1330 : MAGIC SQUARE SOLUTION
1340 : CHECK
1350 :
1360 FOR I=1 TO 4
1370 FOR J=1 TO 4
1380 IF B(I,J)<>M(I,J) GOTO 720
1390 NEXT J
1400 NEXT I
1410 :
1420 : A WIN IS DECLARED!
1430 :
1440 PRINT "THAT IS THE CORRECT SOLUTION!"
1450 INPUT "LIKE TO PLAY ANOTHER GAME",I$
1460 IF LEFT$(I$,1)="Y" THEN RUN
1470 END
```

LISTING 2

```
PROCEDURE numbersquare
BASE 1
(* version 4.0 - 12 NOV 79
(* Marc I. Leavey, MD
(* Basic09 version By Bob Devries April 91
DIM t:INTEGER
DIM i,j:INTEGER
DIM b(4,4):INTEGER
DIM i1,j1:INTEGER
DIM q,m,m9,c:INTEGER
DIM mm(4,4):INTEGER
DIM lop:BOOLEAN
DIM valid:BOOLEAN
DIM solution:BOOLEAN
DIM newgame:BOOLEAN
SHELL "tmode -pause"
newgame=TRUE
WHILE newgame=TRUE DO
  PRINT CHR$(12);
  PRINT "N U M B E R   S Q U A R E S"
  PRINT "---------------------------"
  PRINT
  PRINT "Welcome to the world of"
  PRINT "confusion.  There are two"
  PRINT "versions of number squares:"
  PRINT " 1 - sequential"
  PRINT " 2 - Magic Square"
  INPUT "Which is your pleasure ? ",t
  IF t=1 THEN
    RUN setupssb(b,i1,j1)
  ELSE
    RUN setupmsb(b,mm,i1,j1)
  ENDIF
  PRINT "I am now scrambling the board..."
  FOR q=1 TO 200
    lop=FALSE
    WHILE lop=FALSE DO
      m=1+RND(3)
      IF m=1 THEN
        IF i1<>1 THEN
          b(i1,j1)=b(i1-1,j1)
          b(i1-1,j1)=16
          i1=i1-1
          lop=TRUE
        ENDIF
      ENDIF
      IF m=2 THEN
        IF i1<>4 THEN
          b(i1,j1)=b(i1+1,j1)
          b(i1+1,j1)=16
          i1=i1+1
          lop=TRUE
        ENDIF
      ENDIF
      IF m=3 THEN
        IF j1<>1 THEN
          b(i1,j1)=b(i1,j1-1)
          b(i1,j1-1)=16
          j1=j1-1
          lop=TRUE
        ENDIF
      ENDIF
      IF m=4 THEN
        IF j1<>4 THEN
          b(i1,j1)=b(i1,j1+1)
          b(i1,j1+1)=16
          j1=j1+1
          lop=TRUE
        ENDIF
      ENDIF
    ENDWHILE
  NEXT q
  (* print board line 720
  solution=FALSE
  REPEAT
    m9=0
    RUN gfx2("cwarea",29,12,22,12)
    PRINT "-----------------"
    FOR i=1 TO 4
      FOR j=1 TO 4
        PRINT ": ";
        IF b(i,j)=16 THEN
          PRINT "  ";
        ELSE
          IF b(i,j)<10 THEN
            PRINT " ";
          ENDIF
          PRINT b(i,j);
        ENDIF
      NEXT j
      PRINT ":"
      PRINT "-----------------"
    NEXT i
    (* input move
    valid=FALSE
    WHILE valid=FALSE DO
      i1=0
      j1=0
      WHILE i1=0 DO
        RUN gfx2("bell")
        INPUT "Move which piece ? ",m
        IF m=0 THEN
          RUN gfx2("cwarea",0,0,80,24)
          PRINT CHR$(12)
          SHELL "tmode pause"
          END
        ENDIF
        FOR i=1 TO 4
          FOR j=1 TO 4
            IF b(i,j)=m THEN
              i1=i
              j1=j
            ENDIF
          NEXT j
        NEXT i
        IF i1=0 THEN
          PRINT "I can't find that number"
        ENDIF
      ENDWHILE
      i2=0
      j2=0
      FOR i=i1-1 TO i1+1
        IF i>=1 AND i<=4 THEN
        EXITIF b(i,j1)=16 THEN
          i2=i
          j2=j1
          valid=TRUE
        ENDEXIT
        ENDIF
      NEXT i
      IF valid=FALSE THEN
        FOR j=j1-1 TO j1+1
          IF j>=1 AND j<=4 THEN
          EXITIF b(i1,j)=16 THEN
            i2=i1
            j2=j
            valid=TRUE
          ENDEXIT
          ENDIF
        NEXT j
      ENDIF
      IF valid=FALSE THEN
        m9=m9+1
        PRINT "Not a valid move"
      ENDIF
    ENDWHILE
    b(i2,j2)=m
    b(i1,j1)=16
    IF t=1 THEN
      c=0
      FOR i=1 TO 4
        FOR j=1 TO 4
          IF b(i,j)<c THEN (* reprint board
            solution=FALSE
          ENDIF
          c=b(i,j)
        NEXT j
      NEXT i
      IF solution=TRUE THEN
        PRINT "You got it!"
      ENDIF
    ENDIF
    IF t=2 THEN
      FOR i=1 TO 4
```

```
        FOR j=1 TO 4
          IF b(i,j)<>mm(i,j) THEN (* reprint board
            solution=FALSE
          ENDIF
        NEXT j
      NEXT i
      IF solution=TRUE THEN
        PRINT "That is the correct solution!"
      ENDIF
    ENDIF
  UNTIL solution=TRUE
  INPUT "Like to play another game ? ",i$
  IF LEFT$(i$,1)="n" THEN (* rerun game
    newgame=FALSE
  ENDIF
ENDWHILE
RUN gfx2("cwarea",0,0,80,24)
SHELL "tmode pause"
END
 0DE8

PROCEDURE setupssb
BASE 1
PARAM b(4,4):INTEGER
PARAM i1,j1:INTEGER
DIM i,j:INTEGER
FOR i=1 TO 4
  FOR j=1 TO 4
    b(i,j)=(i-1)*4+j
  NEXT j
NEXT i
i1=4
j1=4
 00F2

PROCEDURE setupmsb
BASE 1
PARAM b(4,4),mm(4,4):INTEGER
PARAM i1,j1:INTEGER
DIM i,j:INTEGER
FOR i=1 TO 4
  FOR j=1 TO 4
    READ mm(i,j)
    b(i,j)=mm(i,j)
  NEXT j
NEXT i
i1=4
j1=2
DATA 1,6,15,8,12,11,2,5,10,13,4,3,7,16,9,14
```

Programming with Solder

This new column will appear from time to time as material is available. Primarily it will consist of hardware projects for the Color Computer, though submissions for other hardware that uses OS-9 are very welcome. Skill level required for assembling these projects will vary with the project.

AT Keyboard to CoCo Interface by Dana Peters

INTRODUCTION

Overview

Want a "real" keyboard for your CoCo? Here are plans for a project to construct an interface to let you use an IBM "AT" keyboard with your CoCo. For those who have re-packaged their hardware inside of a PC case, this is the perfect way of making your machine more useable and professional looking.

The interface connects to the CoCo via the standard keyboard connector. Because it emulates a real CoCo keyboard at the hardware level, no special software drivers are required. This also ensures the AT keyboard will work with all software, even those programs which access the hardware directly (many do).

Besides providing ordinary keyboard facilities, the interface also sports a few extra features. The extra AT keyboard keys can be used as programmable macro keys. The macros can be re-defined at any time using the keyboard, and are stored indefinitely. The ability to reset the computer via the CTRL ALT DELETE key sequence is also provided.

This interface should be usable by all models of CoCo (I, II, and III). Any IBM-compatible "AT" keyboard should work. Please note that the older 84-key XT keyboards are not supported.

Background

There have been many attempts at improving the CoCo keyboard over the years. The original CoCo shipped with what was dubbed a "chicklet" keyboard from the resemblance of the keys in size and texture to the chewing gum by the same name. Universally hated, several companies such as Spectrum Projects, KeyTronic, Micronix, Mark Data, and HJL supplied replacement keyboards which provided a much more professional feel. Tandy's supplied much improved keyboards with the CoCo II and III, but was still molded into the case rather than being detachable. Many CoCo owners (particularly ones who fancied putting their CoCo into a PC case) made their keyboards detachable by grafting a cable between the keyboard and motherboard. There where also attempts at hooking PC keyboards to the CoCo. OWL-Ware produced an AT keyboard interface for the whopping price of $119 US, keyboard not included! (from Rainbow magazine, January 1989, page 69). There is also plans for an interface by Bob Puppo which I have not personally seen. From what I've heard, it works only with XT style keyboards, and is relatively complex.

CONSTRUCTION

Theory of Operation

The central part in this design is the Motorola HC11 microcontroller. It essentially is an 8-bit computer on a chip. There are many versions of the HC11, but the MC68HC711D3 has the following goodies:

* 192 bytes internal RAM
* 4K internal EPROM
* asynchronous and synchronous serial controllers
* 2 general purpose 8-bit I/O ports
* timer system controlling an 8-bit I/O port

The HC11 has many options and abilities I won't go into, but suffice it to say that it is a very capable chip that can "do it all" letting you design circuits with very low chip count. It's a shame this project can't take advantage of all the HC11 has to offer, but with so many diverse options, using them all would be difficult!

To emulate the CoCo keyboard matrix, the Mitel 8808 has been employed. This chip is intended to be used to switch analog telephone calls, but its 8x8 switching matrix is perfect for this application. This could be done with a number of digital chips instead, but the single chip solution allows for a simpler design.

The X25040 is an EEPROM which can provide non-volatile storage for 512 bytes of data. This is used to store programmable macrokey sequences.

Parts

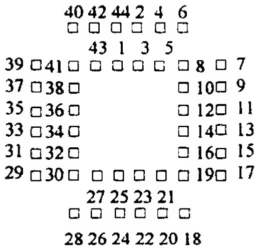
Number	Manufacturer	Package	Description
MC68HC711D3FN	Motorola	48 pin DIP	8 bit microcontroller
MT8808AE	Mitel	28 pin DIP	8x8 analog switch
MC34064	Motorola	TO-92	Low voltage inhibit
X25040	Xicor	8 pin DIP	512 byte EEPROM

The AT keyboard interface requires one of each of the chips in the above table.

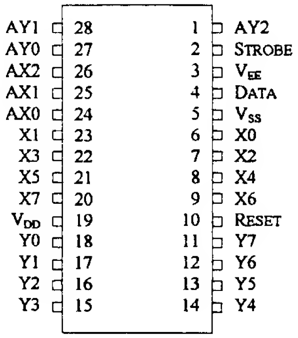
Pinouts

Here are the pinouts of the various components. Please note that these are shown from the bottom of the chips.

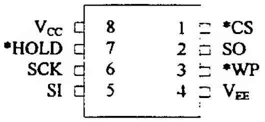
1	EV_{SS}	16	PD0	31	NC
2	V_{SS}	17	PD1	32	PB7
3	PC0	18	MISO	33	PB6
4	PC1	19	MOSI	34	PB5
5	PC2	20	SCK	35	PB4
6	PC3	21	*SS	36	PB3
7	PC4	22	V_{DD}	37	PB2
8	PC5	23	PA7	38	PB1
9	PC6	24	PA6	39	PB0
10	PC7	25	PA5	40	MODB
11	*XIRQ	26	PA4	41	MODA
12	PD7	27	PA3	42	E
13	PD6	28	PA2	43	EXTAL
14	*RESET	29	PA1	44	XTAL
15	*IRQ	30	PA0		



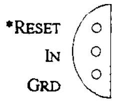
44-PIN PLCC SOCKET (BOTTOM)



MT8808AE



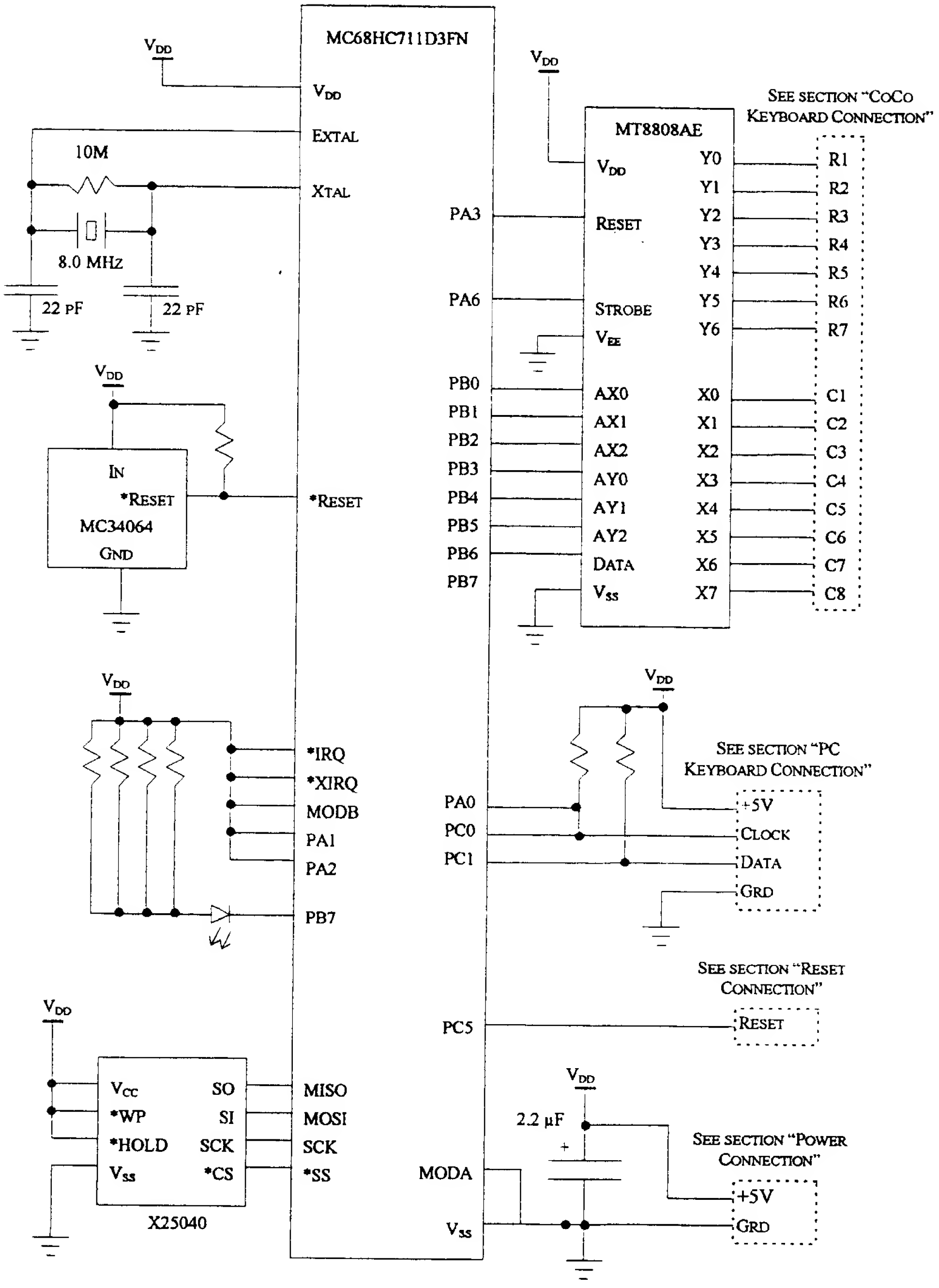
X25040



MC34064

Schematic

All resistors 10K unless otherwise marked.
Pins not on diagram are No Connection.

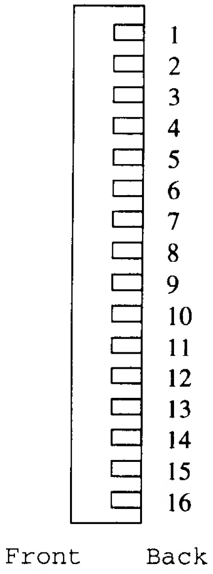


CoCo Keyboard Connection

In this section I assume that you are connecting to an "F" board CoCo I, CoCo II or CoCo III, which all have the mylar film connector for the keyboard. To connect to interface to the computer's keyboard port, I recommend using a row of staking pins which can be plugged into the mylar keyboard connector.

Pin	Desc
1	R1
2	R2
3	NC
4	R3
5	R4
6	R5
7	R6
8	R7
9	C1
10	C2
11	C3
12	C4
13	C5
14	C6
15	C7
16	C8

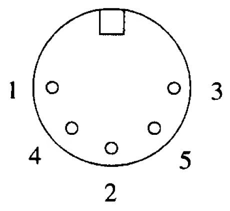
Here is the CoCo's keyboard connector looking down from the top of the board:



PC Keyboard Connection

The PC keyboard connects a 5-pin DIN connector. Radio Shack supplies a cabinet mount connector. Here is the pinout looking at the back of the cabinet mount connector for ease of wiring:

Pin	Desc
1	Clock
2	Data
3	NC
4	Ground
5	+5V



Power Connection

All that the interface and AT keyboard need is +5V; where you get it is up to you. If you have your CoCo inside of a PC case, I suggest taking power from an extra floppy power connector you're likely to have. If not, then steal the power from any convenient place on the CoCo motherboard. Use medium gauge wire from the power supply to the interface board and the interface board to the PC keyboard to avoid a voltage drop. The 2.2æF capacitor should bridge the supply lines where they enter the interface board. I suggest you verify with a voltage meter that what you think you've connected to +5V is actually +5V.

Reset Connection

To allow the CoCo to be reset from the keyboard, a wire must be run to the non-grounded side of the CoCo's reset button. Conveniently, this is available on the top side of the board on most CoCos, right on the reset switch itself.

OPERATION

Keymaps

If you take a close look at both a CoCo and AT keyboard, you will soon realize that there is a distinct difference in the layout of non-alphabetic keys. Because of this, the adapter supports two different modes of operation: PC keymap and CoCo keymap. The two modes can be toggled between using the SCROLL LOCK key. When the SCROLL LOCK LED is off, the adapter is in PC keymap; when it is on, CoCo keymap is in effect.

In PC keymap, the adapter maps the keys based what is printed on the AT keyboard's keycaps. For example, pressing SHIFT 2 will generate the @ character. When in CoCo keymap, pressing the same keys will generate the " character. The CoCo keymap maintains a one-to-one correspondence of PC keys with CoCo keys based partly on their physical layout. This is useful for games which depend on the CoCo's layout of keys, or for users who can't quite teach their fingers a different key layout.

Num Lock

The NUMLOCK key toggles the state of the corresponding keyboard LED, and changes the behavior of the keypad keys. When the NUMLOCK LED is on, the keypad keys behave the same as the keys "0"-"9" and ".". When NUMLOCK is off, the keypad keys behave the same as the arrow keys and macro keys to the keypad's immediate left.

Caps Lock

In PC keymap mode, pressing the CAPSLOCK key toggles the keyboard's caps lock state. While the keyboard's caps lock is on, the adapter will automatically press SHIFT whenever an alphabetic character is typed. Note

that this doesn't affect the computer's caps lock state. Press SHIFT CAPSLOCK (RSDOS) or CTRL 0 (OS9) to toggle the computer's caps lock state.
Note that keyboard caps lock is not effective in CoCo keymap mode. Use SHIFT 0 to toggle RSDOS's caps lock state while in CoCo keymap mode.

Macro Keys

One of the best features of this adapter is it's ability to use the extra keys on the AT keyboard as macro keys. A pre-recorded sequence of keystrokes can be associated with each macro key. The sequence is re-played whenever the key is hit. This is especially useful for commonly used commands, and for defining application specific key combinations for the named keys such as INSERT and PAGE UP. The macro sequences are re-definable at any time through the keyboard, and are saved indefinitely. Each of the 16 available macro keys can have up to 32 keystrokes stored. The sequence is stored exactly as typed, including overlapping keystrokes (one key held down while another is pressed). The timing of keystrokes is not stored, so the keys are replayed at a fixed rate regardless of the delay between keystrokes when recorded.

The macro keys are F1 through F10, and the keys INSERT, DELETE, HOME, END, PAGE UP, and PAGE DOWN. To define a macro key, perform the following steps:

1. Press SHIFT MACROKEY.
2. Type the sequence of characters to be recorded.
3. Press SHIFT MACROKEY again.

The macro keys pressed in the first and third steps must match for the sequence to be stored (as a guard against accidentally re-defining a macro). You cannot use macro key while recording another macro. An attempt to do so will stop the recording. The recording will also stop if the sequence exceeds 32 bytes in length. To erase a macro, simply press SHIFT MACROKEY twice to record a null macro.

To play a macro, press MACROKEY. The keys are played back at a pre-determined rate which cannot be adjusted. The playback is not affected by the state of the keyboard caps lock when invoked; the state of the keyboard caps lock is significant only when the macro is recorded. In contrast, the state of the CoCo caps lock does not affect the recording, but it is significant when the macro is played back.

Note that while recording a macro, it is best to avoid key roll-over in order to reduce the number of bytes needed to store the macro. Roll-over occurs when while typing quickly you press a second key before releasing the first one. Because of how the keys are stored when a macro is recorded, a two key sequence without roll-over takes 2 bytes, where as with roll-over, the same sequence takes 4 bytes. The 32 byte limit is exceeded much more easily if overlapping key sequence information is unnecessarily stored due to roll-over. As a rule of thumb, type slowly when recording macros.

Keyboard Reset

Another desirable feature of the adapter is the ability to reset the CoCo via the "three finger salute," the "Vulcan nerve pinch," or CTRL ALT DELETE key combination. Pressing these three keys will cause the CoCo to be reset.

Pressing SHIFT CTRL ALT DELETE causes the adapter to hold down the CoCo's CTRL and ALT keys while resetting. This is useful for forcing a cold restart on the CoCo III.

NOTES FROM THE AUTHOR

I hope that many people in the CoCo community have as much fun building this interface as I've had designing it. I encourage you to send your questions, comments, and greetings to me via email at the address Dana_Peters@Mitel.COM (use ai745@Freenet.Carleton.CA if that fails for some reason), or by mail:

```
Dana Peters
#7-49 Sherway Drive
Nepean, Ontario
Canada K2J 2K3
```

I had recently completed an HC11 based AT keyboard interface for my Amiga, and thought that it wouldn't be much work to adapt it for use on the CoCo. Enthusiastic feedback from the network lead me to complete the project and produce this document. If you're reading this, you've probably come up with a hardware hack or two on your machine. I'd like to encourage you to document your work (no matter how small), and make it available to the community.

Acknowledgments

Thanks to Alan Dekok for his suggestions and letting me test the interface on his machine, and to David Linsley for providing information regarding the Dragon's keyboard. A few historical tidbits and information regarding the various models of CoCo came from Tandy's Little Wonder, The Color Computer 1980-1991, a book by F.G. Swygert.

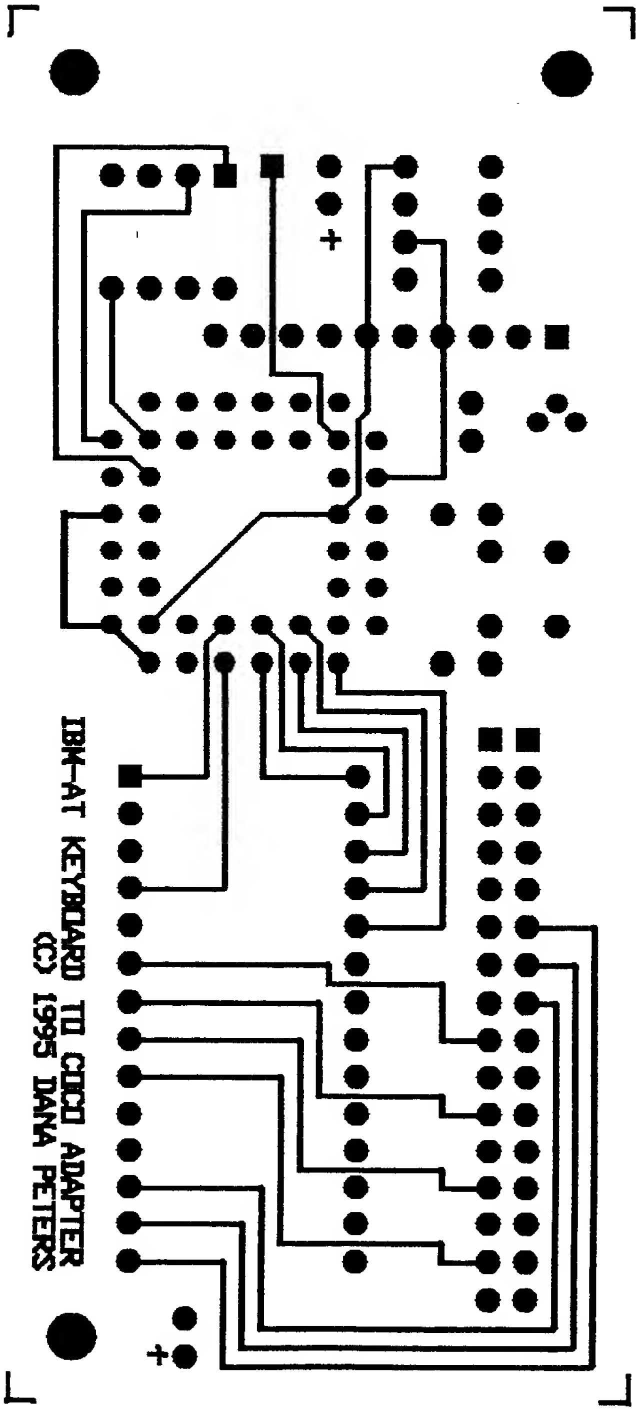
Thanks to Stan Burton, for creating a high quality development system for the HC11 which he was kind enough to release as freeware for the Amiga. ADEV11 is available from Aminet sites.

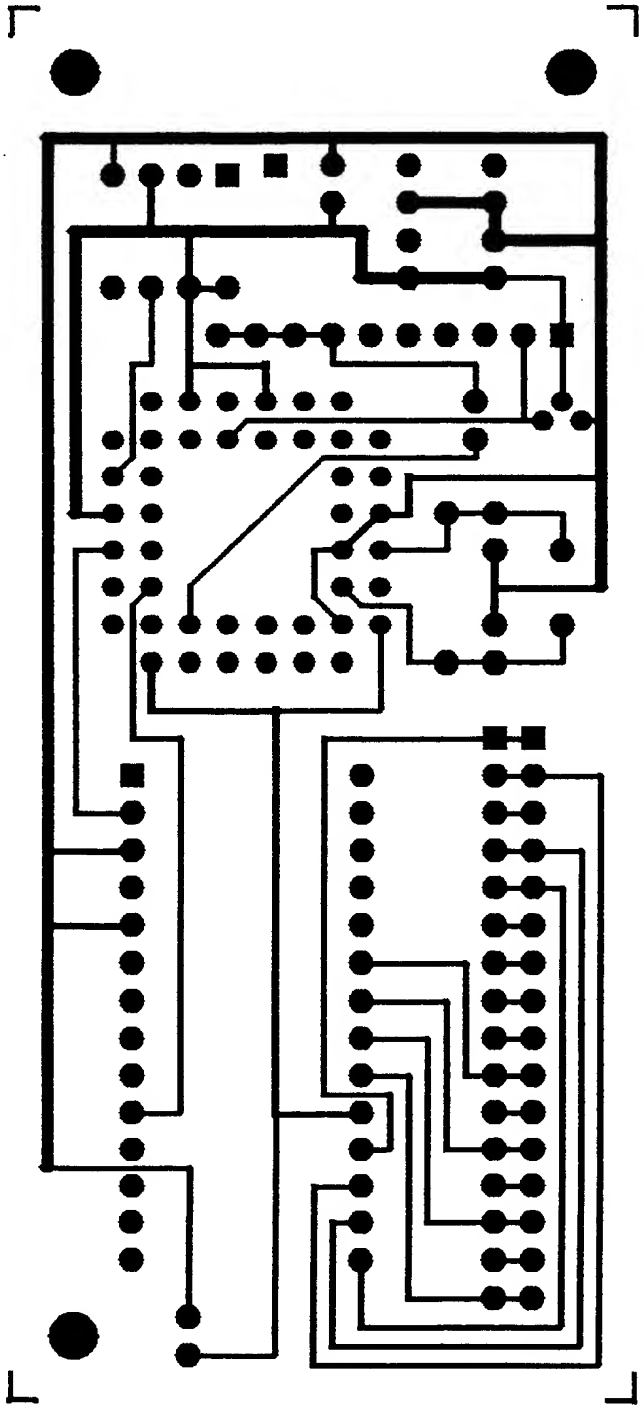
And thank you to my special friend Aysa September for putting up with my geeky projects.

AT KEYBOARD TO COCO INTERFACE (c) 1995 DANA PETERS

Note: Both Kit (PCB, pre-programmed 6811 and other components) and Assembled versions of this project will be available from Northern Xposure beginning in early October. Call for pricing and availability. For those who wish to pursue construction of the project on their own, you can get a 6811 programmed with the software by mailing it to Dana Peters with $5.00 to the above address.

Top and bottom printed circuit board layouts (two-times actual size) appear below.





The OS-9 Users Group Sourcebook

by Chris Perrault

The Users Group is undertaking a project to establish a centralized OS-9 information resource, containing data ranging from OS-9 Users to Bulletin Boards to Vendors. This information will be available to both individual and corporate Users Group members for a nominal charge. The co-ordinator for this project is Chris Perrault.

Vendor Information

In this, our fourth installment of the OS-9 Users' Group Sourcebook series, we will take a look at a major force behind the OS-9 community. Those who sell hardware, software, books, and magazines, related to OS-9, play a very crucial role in our advancement as a legitimate market. If we don't have these vendors, we are effectively non-existent. If you are a company that sells OS-9 related products and services, whether for personal or industrial use, then please fill out and return the form that follows this article.

A Little Background

Since OS-9 first made it's appearance in the personal computing market back in the mid 80's, third party vendor support is what has kept it as a viable system choice. In the present era, the number of companies supporting OS-9 has dwindled but new ones still pop up to replace those that have gone under. The companies that remain are still very active and bring new products to OS-9 users regardless of the fact that the userbase is not the size of many in the more mainstream communities. In order to keep business for the vendors healthy, it is up to US the USERS to support them by continuing to buy their products, and to communicate with them about what you would like to see. Many a company has gone under in the past due to lack of customer participation, and it is the hope of us the OS-9 UG that such participation will increase and users will begin to show their appreciation even more with their patronage.

On The Other End...

There is also an obligation on the part of the companies selling OS-9 related products. In order to continue user confidence and to gain even more of it, you will have to hold up your end of the bargain and meet the users needs on various levels. Getting orders shipped in a timely manner, having the actual product in stock BEFORE taking checks, developing a high standard for your products, along with communicating with your customers on a regular basis, especially in the event that a there may be problem with the product or it's availability are key issues and it is important that vendors adopt a standard they can maintain in addressing them. Traditionally OS-9 vendors have been very good about this in the past, but with such a small scale market to deal with, it does get harder to businesses to meet these needs in the present era. Without a proper balance of support from both the users and the companies, prosperity in this community becomes that much more uncertain, and it doesn't take a lot to throw things off. Through the sourcebook it is our intention to help vendors by keeping users aware of their presence and where to find them, and to also assist in helping vendors find the potential customers. This kind of communication is key in helping to establish a healthy business-community relationship. We ask ALL OS-9 based businesses to fill out and return the supplied entry form, regardless of whether or not you are a UG member. We don't want ANYONE to miss out on this opportunity.

Where Can I Find OS-9 Vendors?

The first place you can look to find OS-9 businesses is right here in this issue of the MOTD. Every issue includes a listing of vendors with addresses and phone numbers, including those that are corporate members supporting the UG. Also many will advertise in the MOTD, plus other publications that support OS-9 such as 'the world of 68 micros', so be sure to subscribe to at least one OS-9 related publication, since they also have some pretty good reading. A list of publications is also included with each issue of the MOTD. Very importantly, coming up on the weekend of September 30 and October 1st will be the Atlanta Fest, where you will be able to meet OS-9 vendors from all over the nation and buy some of their new and not so new products 'live' and in 'real time'. More information regarding the Atlanta Fest is available elsewhere in this issue if you interested are in attending. If you can make the trip you are strongly encouraged to do so and show your encouragement to the vendors and organizations that like us the users help keep OS-9 going!

Entry Level C

by Jim Vestal

Jim Vestal's *Entry Level C* column will return next issue.

The OS-9 Users' Group Sourcebook
Individual User Entry Form

Name: ______________________________

Business: ______________________________

Address: ______________________________

City: ____________ State: __________ ZIP: ____________

Country: ____________ Telephone: (_____) ______-__________

EMail Address: ______________________________

Signature: ______________________ Date: /   /
DD MM YY

Would you like information on joining the OS-9 Users Group? ○ Already a Member! ○ Yes ○ No

Can this information be passed to OS-9 vendors? ○ Yes ○ No

Are you available to act as a volunteer for OS-9 projects from time to time? This isn't a commitment. ○ Yes ○ No

Please indicate your area(s) of interest:

○ OS-9/6809 ○ OS-9/68000 ○ OS-9000

○ Programming ○ Hardware Other ______________

You use OS-9 primarily for what purpose:

○ personal use ○ industrial system ○ both

Please describe briefly your OS-9 computer system(s):

____

____

Please describe briefly your computer and OS-9 experience:

____

____

Please describe your OS-9 interests (things you would like to see the UG do, or see in the MOTD), or list any other questions or comments:

____

____

The OS-9 Users Group Sourcebook
Vendor Entry Form

Company ______________________ Telephone V (_____) _____-_______

Contact Name ______________________ F (_____) _____-_______

Street/ ______________________________

City ______________ State _____ Zip Code ________ Country__________

Email Address ______________________________

Do you support any of the following areas? Please check all that apply.

Software: ○ OS-9000 ○ OS-9/68000 ○ OS-9/6809

Hardware: ○ Complete systems ○ Components ○ Hardware Service/Repair

Windowing System: ○ G-Windows ○ X-Windows ○ K-Windows ○ MGR

Services: ______________________________

What market do you specialize in? ○ Industrial ○ Personal ○ Both

Please tell us a little about your business, including what you specialize in and what some of your products are. (Or attach a catalogue or brochure.)

____

____

____

____

The OS-9 Users Group, Inc.
6158 West 63rd Street
Suite 109
Chicago, Il. 60638
U.S.A.